```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEE              DDD         DDD      TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEEEEEEEEEEE     DDD          DDD     TTT
EEEEEEEEEEEE     DDD          DDD     TTT
EEEEEEEEEEEE     DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEE              DDD          DDD     TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD        TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD        TTT
```

```
LL          FFFFFFFFFF    CCCCCCCC    000000    UU      UU  NN      NN  TTTTTTTTTT
LL          FFFFFFFFFF    CCCCCCCC    000000    UU      UU  NN      NN  TTTTTTTTTT
LL          FF          CC          00      00  UU      UU  NN      NN      TT
LL          FF          CC          00      00  UU      UU  NN      NN      TT
LL          FF          CC          00      00  UU      UU  NNNN    NN      TT
LL          FF          CC          00      00  UU      UU  NNNN    NN      TT
LL          FFFFFFFF    CC          00      00  UU      UU  NN  NN  NN      TT
LL          FFFFFFFF    CC          00      00  UU      UU  NN  NN  NN      TT
LL          FF          CC          00      00  UU      UU  NN    NNNN      TT
LL          FF          CC          00      00  UU      UU  NN    NNNN      TT
LL          FF          CC          00      00  UU      UU  NN      NN      TT          ....
LL          FF          CC          00      00  UU      UU  NN      NN      TT          ....
LLLLLLLLLL  FF            CCCCCCCC    000000    UUUUUUUUUU  NN      NN      TT          ....
LLLLLLLLLL  FF            CCCCCCCC    000000    UUUUUUUUUU  NN      NN      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;    1          0001  0 %TITLE 'EDT$LFCOUNT - type a message with a count'
;    2          0002  0 MODULE EDT$LFCOUNT (                            ! Type a message with a count
;    3          0003  0                     IDENT = 'V04-000'                       ! File: LFCOUNT.BLI Edit: JBS1006
;    4          0004  0                     ) =
;    5          0005  1 BEGIN
;    6          0006  1 !
;    7          0007  1 !**********************************************************************************
;    8          0008  1 !*                                                                                *
;    9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
;   10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                         *
;   11          0011  1 !*  ALL RIGHTS RESERVED.                                                           *
;   12          0012  1 !*                                                                                *
;   13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED         *
;   14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
;   15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER        *
;   16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY        *
;   17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
;   18          0018  1 !*  TRANSFERRED.                                                                   *
;   19          0019  1 !*                                                                                *
;   20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE        *
;   21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT        *
;   22          0022  1 !*  CORPORATION.                                                                   *
;   23          0023  1 !*                                                                                *
;   24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS        *
;   25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                        *
;   26          0026  1 !*                                                                                *
;   27          0027  1 !*                                                                                *
;   28          0028  1 !**********************************************************************************
;   29          0029  1 !
;   30          0030  1
;   31          0031  1 !++
;   32          0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
;   33          0033  1 !
;   34          0034  1 ! ABSTRACT:
;   35          0035  1 !
;   36          0036  1 !       This module writes out that portion of a line mode message
;   37          0037  1 !       giving a count.
;   38          0038  1 !
;   39          0039  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
;   40          0040  1 !
;   41          0041  1 ! AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
;   42          0042  1 !
;   43          0043  1 ! MODIFIED BY:
;   44          0044  1 !
;   45          0045  1 ! 1-001 - Original.  DJS 02-FEB-1981.  This module was created by
;   46          0046  1 !       extracting the routine EDT$$FMT_STRCNT  from the module EXEC.BLI.
;   47          0047  1 ! 1-002 - Regularize headers.  JBS 19-Mar-1981
;   48          0048  1 ! 1-003 - Make it work for 32 or 48 bits so pass count pointer.  SMB 5-Feb-1982
;   49          0049  1 ! 1-004 - Change "division" to a routine call.  SMB 11-Feb-1982
;   50          0050  1 ! 1-005 - Modify to use new compare macro. STS 20-Oct-1982
;   51          0051  1 ! 1-006 - Improve the appearance of the listing.  JBS 14-Jun-1983
;   52          0052  1 !--
;   53          0053  1
```

```
;    55          0054  1 %SBTTL 'Declarations'
;    56          0055  1 !
;    57          0056  1 ! TABLE OF CONTENTS:
;    58          0057  1 !
;    59          0058  1
;    60          0059  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    61          0498  1
;    62          0499  1 FORWARD ROUTINE
;    63          0500  1     EDT$$FMT_STRCNT : NOVALUE;                  ! Format a count for printing
;    64          0501  1
;    65          0502  1 !
;    66          0503  1 ! INCLUDE FILES:
;    67          0504  1 !
;    68          0505  1
;    69          0506  1 REQUIRE 'EDTSRC:EDTREQ';
;    70          0641  1
;    71          0642  1 !
;    72          0643  1 ! MACROS:
;    73          0644  1 !
;    74          0645  1 !       NONE
;    75          0646  1 !
;    76          0647  1 ! EQUATED SYMBOLS:
;    77          0648  1 !
;    78          0649  1 !       NONE
;    79          0650  1 !
;    80          0651  1 ! OWN STORAGE:
;    81          0652  1 !
;    82          0653  1 !       NONE
;    83          0654  1 !
;    84          0655  1 ! EXTERNAL REFERENCES:
;    85          0656  1 !
;    86          0657  1 !       In the routine
```

```
;    88          0658  1 %SBTTL 'EDT$$FMT_STRCNT  - type a message with a count'
;    89          0659  1
;    90          0660  1 GLOBAL ROUTINE EDT$$FMT_STRCNT (                    ! Type a message with a count
;    91          0661  1     N,                                               ! The count
;    92          0662  1     S,                                               ! Pointer to message
;    93          0663  1     L                                                ! Length of message
;    94          0664  1     ) : NOVALUE =
;    95          0665  1
;    96          0666  1 !++
;    97          0667  1 ! FUNCTIONAL DESCRIPTION:
;    98          0668  1 !
;    99          0669  1 !       This routine writes out the portion of a message giving a count.
;   100          0670  1 !       The count can be 48 bits long or less
;   101          0671  1 !
;   102          0672  1 ! FORMAL PARAMETERS:
;   103          0673  1 !
;   104          0674  1 !  N                    the count pointer, which is written as a decimal number unless it
;   105          0675  1 !                       is zero, in which case it is written as 'No'
;   106          0676  1 !
;   107          0677  1 !  S                    a pointer to a string of characters which is written after the count,
;   108          0678  1 !                       followed by an 's' unless the count is exactly 1.
;   109          0679  1 !
;   110          0680  1 !  L                    the length of the sting pointed to by S.
;   111          0681  1 !
;   112          0682  1 ! IMPLICIT INPUTS:
;   113          0683  1 !
;   114          0684  1 !       EDT$$L_LNO_ZERO
;   115          0685  1 !       EDT$$L_LNOO-14
;   116          0686  1 !
;   117          0687  1 ! IMPLICIT OUTPUTS:
;   118          0688  1 !
;   119          0689  1 !       NONE
;   120          0690  1 !
;   121          0691  1 ! ROUTINE VALUE:
;   122          0692  1 !
;   123          0693  1 !       NONE
;   124          0694  1 !
;   125          0695  1 ! SIDE EFFECTS:
;   126          0696  1 !
;   127          0697  1 !       NONE
;   128          0698  1 !
;   129          0699  1 !--
;   130          0700  1
;   131          0701  2     BEGIN
;   132          0702  2
;   133          0703  2     EXTERNAL ROUTINE
;   134          0704  2         EDT$$LDIV,
;   135          0705  2         EDT$$FMT_CH,
;   136          0706  2         EDT$$FMT_DCML,
;   137          0707  2         EDT$$FMT_STR;
;   138          0708  2
;   139          0709  2     EXTERNAL
;   140          0710  2         EDT$$L_LNOO : LNOVECTOR [14],
;   141          0711  2         EDT$$L_LNO_ZERO : LN_BLOCK;
;   142          0712  2
;   143          0713  2     LOCAL
;   144          0714  2         DIGIT,
```

```
;  145          0715  2          LINNO : LN_BLOCK,
;  146          0716  2          SIGNIF;
;  147          0717  2
;  148          0718  2      !+
;  149          0719  2      ! Fetch the integer into a local
;  150          0720  2      !-
;  151          0721  2          MOVELINE (.N, LINNO);
;  152          0722  2
;  153          0723  3          IF (LINNOEQL (EDT$$L_LNO_ZERO, LINNO))
;  154          0724  2          THEN
;  155          0725  2              EDT$$FMT_STR (UPLIT ('No'), 2)
;  156          0726  2          ELSE
;  157          0727  3              BEGIN
;  158          0728  3              SIGNIF = 0;
;  159          0729  3      !+
;  160          0730  3      ! Loop once for each possible digit in the number starting with most
;  161          0731  3      ! significant
;  162          0732  3      !-
;  163          0733  3
;  164          0734  3              DECR I FROM 14 TO 0 DO
;  165          0735  4                  BEGIN
;  166          0736  4                  EDT$$LDIV (LINNO, DIGIT, .I);
;  167          0737  4      !+
;  168          0738  4      ! Write the digit out if the current digit is non-zero or
;  169          0739  4      ! we have seen a previous non zero digit
;  170          0740  4      !-
;  171          0741  4
;  172          0742  5                  IF ((.DIGIT NEQ 0) OR (.SIGNIF NEQ 0))
;  173          0743  4                  THEN
;  174          0744  5                      BEGIN
;  175          0745  5                      EDT$$FMT_CH (.DIGIT + %C'0');
;  176          0746  5                      SIGNIF = .SIGNIF + 1;
;  177          0747  4                      END;
;  178          0748  4
;  179          0749  3                  END;
;  180          0750  3
;  181          0751  2              END;
;  182          0752  2
;  183          0753  3          IF (.L NEQ 0)
;  184          0754  2          THEN
;  185          0755  3              BEGIN
;  186          0756  3              EDT$$FMT_STR (.S, .L);
;  187          0757  3
;  188          0758  3              IF ( NOT LINNOEQL (.N, EDT$$L_LNO0)) THEN EDT$$FMT_CH (%C's');
;  189          0759  3
;  190          0760  2              END;
;  191          0761  2
;  192          0762  1      END;                                    ! of routine EDT$$FMT_SCRCNT

                                                                     .TITLE  EDT$LFCOUNT EDT$LFCOUNT - type a message with a
;                                                                                        count
                                                                     .IDENT  \V04-000\
                                                                     .PSECT  _EDT$CODE,NOWRT,  SHR,  PIC,2

                                 00  00  6F  4E  00000 P.AAA:      .ASCII  \No\<0><0>                                    ;
```

```
                                                                .EXTRN  EDT$$LDIV, EDT$$FMT_CH
                                                                .EXTRN  EDT$$FMT_DCML, EDT$$FMT_STR
                                                                .EXTRN  EDT$$L_LNO0, EDT$$L_LNO_ZERO

                                                00FC 00000      .ENTRY  EDT$$FMT_STRCNT, Save R2,R3,R4,R5,R6,R7       ; 0660
                                        57 00000000G    00  9E 00002    MOVAB   EDT$$FMT_CH, R7
                                        56 00000000G    00  9E 00009    MOVAB   EDT$$FMT_STR, R6                      ;
                                        5E              0C  C2 00010    SUBL2   #12, SP                               ;
                                04  AE      04          BC  28 00013    MOVC3   #6, @N, LINNO                         ; 0721
                                        04  AE 00000000G    00  D1 00019    CMPL    LOW_1, LOW_2                  ; 0723
                                                            14  12 00021    BNEQ    1$                            ;
                                        08  AE 00000000G    00  B1 00023    CMPW    HIGH_1, HIGH_2                ;
                                                            0A  12 0002B    BNEQ    1$                            ;
                                                            02  DD 0002D    PUSHL   #2                            ; 0725
                                                    CA      AF  9F 0002F    PUSHAB  P.AAA                         ;
                                        66                  02  FB 00032    CALLS   #2, EDT$$FMT_STR              ;
                                                            26  11 00035    BRB     5$                            ;
                                        52                  0E  7D 00037 1$:    MOVQ    #14, I                    ; 0734
                                                            52  DD 0003A 2$:    PUSHL   I                         ; 0736
                                                    04      AE  9F 0003C    PUSHAB  DIGIT                         ;
                                                    0C      AE  9F 0003F    PUSHAB  LINNO                         ;
                                    00000000G       00          03  FB 00042    CALLS   #3, EDT$$LDIV             ;
                                                            6E  D5 00049    TSTL    DIGIT                         ; 0742
                                                            04  12 0004B    BNEQ    3$                            ;
                                                            53  D5 0004D    TSTL    SIGNIF                        ;
                                                            09  13 0004F    BEQL    4$                            ;
                                7E                  6E          30  C1 00051 3$:    ADDL3   #48, DIGIT, -(SP)     ; 0745
                                                    67          01  FB 00055    CALLS   #1, EDT$$FMT_CH           ;
                                                            53  D6 00058    INCL    SIGNIF                        ; 0746
                                        DD                  52  F4 0005A 4$:    SOBGEQ  I, 2$                 ; 0734
                                                    0C      AC  D5 0005D 5$:    TSTL    L                         ; 0753
                                                            26  13 00060    BEQL    7$                            ;
                                                    7E      08  AC  7D 00062    MOVQ    S, -(SP)                  ; 0756
                                                    66          02  FB 00066    CALLS   #2, EDT$$FMT_STR          ;
                                    50      04      AC          04  C1 00069    ADDL3   #4, N, R0                 ; 0758
                                        00000000G   00      04  BC  D1 0006E    CMPL    @N, LOW_2                 ;
                                                            09  12 00076    BNEQ    6$                            ;
                                        00000000G   00          60  B1 00078    CMPW    (R0), HIGH_2              ;
                                                            07  13 0007F    BEQL    7$                            ;
                                                    7E      73  8F  9A 00081 6$:    MOVZBL  #115, -(SP)           ;
                                                    67          01  FB 00085    CALLS   #1, EDT$$FMT_CH           ;
                                                                04  00088 7$:    RET                               ; 0762

; Routine Size:  137 bytes,    Routine Base:  _EDT$CODE + 0004

;  193          0763  1
;  194          0764  1 !<BLF/PAGE>
```

```
;  196          0765  1 END                                          ! of module EDT$LFCOUNT
;  197          0766  1
;  198          0767  0 ELUDOM



;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _EDT$CODE                        141  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                                Library Statistics
;
;                                          --------- Symbols ---------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped       Time
;
;  _$255$DUA28:[EDT.SRC]EDT.L32;1            377         8         2          40        00:00.2
;  _$255$DUA28:[EDT.SRC]PSECTS.L32;1           2         1        50           7        00:00.1



;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:LFCOUNT/OBJ=OBJ$:LFCOUNT MSRC$:LFCOUNT.BLI/UPDATE=(ENH$:LFCOUNT
;       )

; Size:         137 code + 4 data bytes
; Run Time:          00:12.7
; Elapsed Time:      00:16.2
; Lines/CPU Min:      3635
; Lexemes/CPU-Min: 12199
; Memory Used:  90 pages
; Compilation Complete
```

KEYDEFKEY
LIS

KEYFMTSTR
LIS

KEYCHR
LIS

KEYIMMINP
LIS

KEYCOM
LIS

KEYPADDEF
LIS

KEYPUTCHR
LIS

LCLEAR
LIS

LDIVISION
LIS

LDEFK
LIS

LDELETE
LIS

LFILL
LIS

LRECOUNT
LIS

KEYPAD
LIS

KEYTRNCHR
LIS

LDEFM
LIS

LFLNO
LIS